

Note: This document is also available in Section 6 “Distributed Consensus” of <https://download.wpsoftware.net/bitcoin/alts.pdf>, which provides further context for distributed consensus problems.

1 Introduction

Proof-of-stake is frequently proposed as a mechanism for distributed consensus in non-Bitcoin cryptocurrencies (“altcoins”). However, this idea appears to be fundamentally flawed. We explore the history and motivation leading to Bitcoin’s distributed consensus mechanism, which evades a impossibility result, and demonstrate that proof of stake does not work as a replacement.

2 Proof of Stake

1. What is Proof of Stake?

With the advent of modern cryptography, the idea that *information* can be physically real — and valuable — has moved from the dingy halls of philosophy departments to the concrete world of business. We are all familiar with the economic activity enabled by secure communication: negotiations, contracts, transactions, sales and commands can be sent on the public Internet with no fear of forgery or interception. We are also familiar with the financial consequences when secret data is lost or stolen.

Since the advent of cryptographic currency in January 2009 [3] this notion of valuable information has been made concrete. It is possible to hold and exchange a *fungible* store of value, using public communication media, with cryptographic rather than physical security preventing fraud or theft. Rather than saying “this encryption key is worth \$10,000 because that’s what it will cost us if its encrypted data is exposed” one can say “this key is worth \$10,000 but can be broken up, sending only \$20 of it to another party while keeping the rest”.

With this context, proof-of-stake is a simple idea. A proof of stake is a cryptographic proof of ownership. With cryptocurrencies, it is possible for a proof-of-stake to not only prove ownership of a precise amount of currency, but also prove that this currency satisfies some property (say, it is locked and unspendable until some contract is satisfied).

In particular, proven stake in a scarce and experimental cryptocurrency can be considered a proof of vested interest in the project’s success. By proving stake which is time-locked, it can be used to prove interest in the project’s continued (and sustainable) existence.

2. What is distributed consensus?

A *distributed consensus*, as the term is used in Bitcoin, is a consensus (i.e. global agreement) between mutually-distrusting parties who lack identities and were not necessarily present at the time of system set up. We do allow and require the existence of a synchronous network,

i.e. a network in which all valid data reaches all parties in a reasonable amount of time. We do not (and cannot, in an untrusted and physically dispersed network) assume that nodes agree on the precise timing or even time-ordering of messages on the network.

For the purposes of cryptocurrency, it is sufficient to achieve distributed consensus on the time-ordering of transactions (and nothing else). This implies consensus on the “first transaction which moves these particular funds”, which assures the funds’ new owner that the network recognizes them as such.

The reason that this consensus is needed is called the *double-spending problem*. That is, in any decentralized digital currency scheme there is the possibility that a spender might send the same money to two different people, and both spends would appear to be valid. Recipients therefore need a way to be assured that there are no conflicts, or that if there are conflicts, that the network will recognize their version as the correct one. A distributed consensus on transaction ordering achieves this: in the case of conflict, everyone agrees that the transaction which came first is valid while all others are not.

(The other problems with digital currency, e.g. authentication and prevention of forgery, are comparatively easy and can be handled with traditional cryptography.)

3. How does Bitcoin achieve distributed consensus?

It can be mathematically proven that given only a synchronous network it is impossible to achieve distributed consensus in a cryptographically guaranteed way [1]. Bitcoin achieves the impossible by weakening its requirement from cryptographic guarantee to a mere economic one. That is, it introduces an opportunity cost from outside of the system (expenditure on computing time and energy) and provides rewards within the system, but only if consensus on an unbroken transaction history is maintained.

To accomplish this, Bitcoin provides a way to prove, for each candidate history, (a) that opportunity cost was forfeited, and (b) how much. This is a so-called *proof-of-work*. Furthermore, the work proven includes that of all participants who worked on the history¹ [2]. The consensus history is the one with most total work (at least as far as it has propagated through the network — our weak synchronicity requirement means that the consensus on the most recent part of the history is uncertain). Since the consensus history is the only one containing spendable rewards for work done, this means (a) that provers have an incentive to work on the same history that other provers are, and (b) individual provers can’t take control of the history because they need their peers’ contribution.

4. How is proof of stake used to achieve distributed consensus?

Essentially, the idea behind using proof-of-stake as a consensus mechanism is to move the opportunity costs from outside the system to inside the system. The motivation for this is that using “most proven work” as a criteria for consensus creates an economic incentive to prove as much work as possible. For Bitcoin, which proves thermodynamic work (i.e. a certain

¹In particular, the work done even by miners who don’t find blocks is included, in exactly the same sense that gas molecules in a box contribute to its ambient temperature even if they don’t happen to collide with the thermometer during measurement. This is not an analogy. The principles are the same.

amount of irreversible computation was done), there is a physical limit — the Landauer limit — which controls what “as much work as possible” means². The result of this limit is a consensus which is extremely resource-intensive, producing entropy and driving us toward the heat death of the universe literally as fast as the laws of physics will allow³. By moving the opportunity costs into a human-designed cryptocurrency, it should be possible to construct laws which force much smaller limits on resource consumption.

On a lower level, the way that proof of stake works is that currency holders are able to lock their currency for some amount of time, renting “stake” which is cryptographically verifiable. Then to extend the consensus history, rather than attaching a proof of work, each stakeholder digitally signs the extension. For reasons of practicality, typically a small random selection of stakeholders is chosen for each extension, and only a majority of the selection are required to give the extension validity. The chosen stakeholders are given a reward and after some time they are able to unlock their stake if they so desire.

The idea is that rather than depending on the economic inviability of taking control of a history, stakeholders are incentivized to agree on each extension because (a) they are randomly chosen and therefore unlikely to be in collusion, and (b) even if they can collude, they do not want to undermine the system (e.g. by signing many conflicting histories) because they want to recover their stored value when their stake comes unlocked, and (c) they have limited capacity to cause havoc anyway, since for the above reasons the next random selection of stakeholders will probably choose only a single reasonable history to extend.

5. What is wrong with this mechanism for consensus?

On a high level, by tying our stake to (temporarily) sacrificed cryptographic resources, we are begging the question of consensus on who is in possession of what resources. Proof of stake advocates attempt to evade this accusation by pointing out that false histories can only be created by stakeholders, and their power is limited to a short interval of time (the time when they are the chosen signers) during which they are incentivized not to do so. Therefore conflicting histories simply will not appear, and we can appeal to synchronicity of the network to obtain consensus on the one existing history.

The problem with this argument is simple: the “short interval of time” is only short as measured by the consensus history, which only corresponds to a short interval in real time if there exists a consensus history. So we are still begging the question. In fact, if a stakeholder later irreversibly sells his stake for some resource outside the system (e.g. at an exchange), he no longer has incentive not to fork the history (or worse, expose his keys and let others fork the

²This is why we consider the proof of work to be a “proof”, by the way: as long as our hash function is strong, the laws of physics prevent cheating.

³As an aside, it is interesting to note that rather than using a proof-of-work limited by the thermodynamic limit of computations per second, it should be possible to construct a proof-of-work which is limited by the bandwidth of the universe, i.e., the uncertainty principle which puts a lower bound on the size of information storage along with the speed of light which puts bound on how fast information can travel from storage to storage. Since information transfer is reversible, the resultant proof of work should require large amounts of entropy production. This is the premise behind a *memory-hard* proof-of-work, which is outside the scope of this article. See for example [4]. There are many subtleties to this but the main concern with such a proof-of-work is that it shifts proving costs from marginal expenses to capital ones, which for a currency may cause economic incentives toward an oligarchy.

history) at the point in consensus time when he had control.

This is a bit abstruse. We can illustrate it with an example. Suppose that at some early point in consensus time, a single person has the ability to extend history. (For example, they have control over every key which a new block is required to be signed by.) This may have happened organically, if this person's keys were chosen randomly by the stake-choosing algorithm, but it could also happen if this person tracks down the other keyholders and buys their keys. This may happen much later in consensus time (and real time), so there is no reason to believe these keyholders are still incentivized to keep their keys secret. Alternately, they may have revealed the keys through some honest mistake, the chances of which increase as time passes, backups are lost, etc.

Now, we have a consensus history and an attacker who is able to fork it at some early time. To actually replace the entire consensus history, he needs to produce an alternate history, starting from his fork, which is longer than the existing history. But every block needs a new random selection of signers, so is this possible? The answer is absolutely yes: we have been using this word "random", but in fact we have required consensus on the set of signers (otherwise forks would trivially happen), so even a random selection must be seeded from past consensus history. Therefore, an attacker with enough past signing keys can modify the history he has direct control over, causing future signer selections to always happen in his favour. (It is likely he needs to "grind" through many choices of block before he finds one which lets him keep control of the signer selection. In effect, he has replaced proof-of-stake with proof-of-work, but a centralized one.)

Further, this ability to control the future selection of stakeholders (and even the *set* of stakeholders, by controlling which transactions appear in blocks) has serious consequences. This is because even without a deliberate attacker, the signers who extend the history at every point have an incentive to direct the history toward one in which they have more stake (and therefore more reward), which causes the system to trend toward centralization. They may do this by skewing the stake selection of future blocks, or more insidiously by censoring transactions which (may eventually) increase the set of stakeholders.

6. Is it possible to obtain a distributed consensus without provably consuming some resource outside of the system?

Intuitively, the answer is no, but there is no rigorous argument for this claim.

The problem ultimately comes down to what Greg Maxwell calls *costless simulation*, and Andrew Miller calls *nothing at stake*. If it is costless for signers to create valid blocks, then they are able to cheaply search the blockspace for blocks which direct the history in their favour. No matter how the network is designed to prevent a minority takeover, an attacker can direct history toward a present in which they *are* the majority, as determined by the consensus, even if they are only a single party in physical space.

It would therefore appear that whatever space we want to achieve distributed consensus in (in Bitcoin's case, it is the space of humans, which can we approximate by thermodynamic

space since we are autonomous agents within that space), we need to consume resources *in that space* to get the consensus.

3 About this document.

This eventually should be merged into the “Distributed Consensus” section of *A Treatise on Alt-coins*, but as that is very unfinished, in the meantime the author is releasing it as a standalone article.

Because of the interaction between incentive analysis and cryptography, this subject makes for frequent and mentally-taxing discussions on IRC. The goal of this document is to clarify the fundamentals and put these discussions on sounder footing.

References

- [1] M.A. Fischer, N.A. Lynch, and M.S. Paterson, *Impossibility of distributed consensus with one faulty process*, J. Ass. for Comp. Mach. **32** (1985), no. 2, 374–382.
- [2] A. Miller, E. Shi, A. Juels, B. Parno, and Katz. J., *Permacoin: Repurposing bitcoin work for data preservation*, Proceedings of the IEEE Symposium on Security and Privacy, IEEE, 2014, <http://cs.umd.edu/~amiller/permacoin.pdf>.
- [3] S. Nakamoto, *Bitcoin: A peer-to-peer electronic cash system*, 2009, <https://www.bitcoin.org/bitcoin.pdf>.
- [4] J. Tromp, *Cuckoo cycle: a memory-hard proof-of-work system*, <https://github.com/tromp/cuckoo>, 2014.